1-1-1980

Interactive computer graphics animation systems.

Suzanne Davis Vitale

INTERACTIVE
COMPUTER GRAPHICS ANIMATION SYSTEMS

by

Suzanne Davis Vitale

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Computer Science

Lehigh University

1980

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

March 20 1980
(date)

[Gerhard Rayna]
Professor in Charge

[Gilbert Stengle]
Chairman of Department

TABLE OF CONTENTS

		page
	Abstract	1
I	Introduction	2
II	Specific definition of ENANIM	5
III	Optimum system design	6
IV	Hardware and software involved	16
V	Actual system design	17
VI	Comparison with existing animation systems	24
VII	Evaluation of system	35
VIII	Bibliography	39
IX	Vita	40

ABSTRACT

An interactive computer graphics animation system is both an art form and a science form. Instead of creating animation by hand, drawing frame after frame, it is possible to use a computer to do the animation. The use of the computer to create movement in real-time is an invaluable tool used for many purposes. This thesis describes one such animation system called ENANIM, which was created for visual aid in interpreting analytical results obtained from engineering analysis programs. Besides developing an animation system, several other animation systems were studied and compared to ENANIM. Although the other systems were not identical to ENANIM in design or purpose a good comparison could be drawn between the various systems. Before developing ENANIM an optimum animation system was designed. This was done without regard to computer hardware or software restrictions. The only thing taken into consideration was the purpose of the package. A comparison was made between the optimum system design and the actual system design. One conclusion drawn here was that the actual system is more restrictive than desired. This could not be helped because of the available software and hardware. Because of the differences between the various animation systems, ENANIM had to be evaluated on its own merit. Given the purpose of ENANIM, that of visual aid, it does function as it should. It can be of use to an engineer who wants to have a better understanding of some dynamic process that is hard to visualize when seeing the output on paper.

INTRODUCTION

"Animation is the graphic art which occurs in time. Whereas a static image (such as a Picasso or a complex graph) may convey complex information through a single picture, animation conveys equivalently complex information through a sequence of images seen in time. It is characteristic of this medium, as opposed to static imagery, that the actual graphical information at any given instant is relatively slight. The source of information for the viewer of animation is implicit in picture change: change in relative position, shape, and dynamics. Therefore, a computer is ideally suited to making animation 'possible' through the fluid refinement of these changes."[1]

The word animation has different connotations to different people. Most people immediately think of cartoon characters such as Mickey Mouse and Donald Duck. In this paper the term "animation" is used as is described in the above quotation. It is movement in time. Add the phrase "interactive computer graphics" to the word animation and you have a system which uses a computer to produce a sequence of movements in real-time.

The use of computers for producing animation packages is relatively new. The past fifteen years have seen the entrance and development of graphics in the computer world. This, of course, was dependent on the development of both computer hardware and software. In the early and mid 1960's computer companies such as DEC, IBM and Tektronix developed the necessary hardware, and industries such as General Motors and Lockheed were pioneers in designing large scale

---

1. R.M. Baecker, "Picture-Driven Animation", AFIPS, Vol. 34, (1969), p. 273.

systems software and applications programs for the new hardware. At M.I.T. Lincoln Laboratories, I.E. Sutherland and his associates introduced and popularized many fundamental notions of graphics still in use today.[2]

Once the necessary hardware and software was developed, ideas for applications of computer graphics mushroomed. Producing animated films was going on long before computer graphics was developed, but here was an ideal computer application. Many systems have been developed in the past fifteen years that use a computer system to produce an animated film (i.e. MOGUL at Johns Hopkins University, GENESYS at M.I.T.). Another application for computer animation is for educational purposes. An example is a system called ANTICS, which was developed at the University of British Columbia to animate LISP programs. A student could sit in front of a CRT and interactively communicate with ANTICS to help in learning the computer language LISP. A third application for computer animation is for visual aid. No other medium allows a person to view something changing with time as easily as a computer. Instead of seeing a lot of numbers or static pictures on paper and trying to discern how something changes with time, computer animation lets the viewer see the results actually move in time. "Movement is represented as it is perceived, as (potentially) continuous flow, rather than as a series of intermed-

---

2. James D. Foley and Andries van Dam, Interactive Computer Graphics, Addison-Wesley, 1979.

iate states."[3] The applications for computer animation are endless. Consequently this area is rapidly developing into an important section of computer science.

---

3. R.M. Baecker, "Picture-Driven Animation", AFIPS, Vol. 34, (1969), p. 274.

SPECIFIC DEFINITION OF ENANIM

Very often an engineer will run an analysis of a problem and get the results printed or plotted on paper. These results are mostly time dependent because the problems are usually dynamic in nature. The static representation of problems that change with time is sometimes hard to visualize. As an outgrowth of the above problem this thesis was developed. Here is an area where a computer graphics animation system would be of great value. The system produced is called ENANIM - ENgineering ANIMation system. It is used as a visual aid by engineers to help in the interpretation of analytical results produced from dynamic problems.

ENANIM is an interactive system. The computer asks the user questions and the user responds. From the responses given the system produces the results of some analysis moving in real-time. This system was designed for the engineer who has little if any computer graphics experience. As was stated above this package is a visual aid for interpreting results. The engineer must first analyze and run the problem involved. From that point on the computer animation system ENANIM takes control to display the results. This system will allow the user to not only view the results move in real-time, but also alter certain aspects of the display, such as scale and screen organization, to better aid in interpretation. ENANIM was designed so an engineer could have a better understanding of the processes occuring in a wide range of dynamic problems.

OPTIMUM SYSTEM DESIGN

When developing a new computer system the foremost thought in the designer's mind is to create the optimum design for the particular system needed. To design the perfect system, one must put aside the reality of available equipment and just consider what would be the best configuration of a system given unlimited resources. Although this is an unrealistic way to go about designing something it was done here to bring out all the best possible options that could be incorporated in a computer graphics animation system. To be more specific, the following question is posed: What must be considered when developing an interactive computer graphics animation package to be used by engineers for visual aid in interpreting results from engineering problems? Some of the problems to be resolved are the following: What type of interaction will the user have with the computer? What options should be offered to the user? How and where should the pictures appear on the screen?

The system design described below is an optimum system design. When the availability of equipment is brought back into perspective this design changes. The animation system actually implemented at Bethlehem Steel Corporation, Homer Research Labs is described fully in a later chapter. Then the optimum system conceived is compared to the actual system implemented.

This graphics system is to be interactive. Therefore an important design question is what type of interaction will there be between the user and the computer. This is a one person, one machine interaction. The user will sit down in front of a CRT and type on the keyboard the appropriate words to enter the animation package. (In the case of the actual system which was implemented these are 'RUN ENANIM'). The interaction between user and machine consists of questions printed on the screen and the user making appropriate responses. In the ideal situation the user would be able to input her responses via the keyboard, the function keys and the joyswitch. When directly answering the questions posed by the computer the user would use the keyboard for her responses. For other possible action the function keys, joyswitch and special keys would be used. For example, if the user is viewing a model and wishes to stop the action of the picture, all she has to do is depress a predetermined function key, say 'S'. To continue the motion of the picture after having suspended the action the user would hit the 'C' key. To erase the screen at any time the user would hit the 'E' key. One of the options that will be discussed later is the ability to change the speed of motion of the picture. How is this best accomplished? The use of the joyswitch is an ideal method for controlling speed. If the user wants to speed up the action, all she has to do is push the joyswitch forward (away from herself). In order to slow down the motion of the picture, the user can pull the joyswitch back (toward herself). When the split screen option (described later) is being utilized in this system the controls will alter slightly. The user will be given the ability to control each model separately. This will be done with the use of the function keys. The right six function keys will control the right model and the left six function keys will control the left model. The joyswitch speed control will work on both models. If a different speed for each model is desired the function keys would be used. This type of interaction allows the user flexibility and ease of working with this system.

This system is using a CRT as its display. What is displayed on the screen is an important question. All text that appears on the screen will be displayed in storage. This includes the menu that will list all the possible options allowed the user. When something is displayed on the screen in storage, it is drawn once and remains visible until the screen in erased. Pictures, on the other hand, will be displayed in refresh. Something displayed in refresh is not "stored" on the screen. It must be continually drawn and redrawn on the screen to remain visible. Pictures are mostly drawn in refresh because of the action involved. To have the illusion of movement objects must be in refresh.

While viewing a model the user will probably want to know the amount of time that has elapsed. Therefore, besides the model being displayed in the middle of the screen, there will be a clock in the upper right-hand corner advancing in time as the model moves with time. This time clock will correspond to the time values received from the engineering analysis results.

The possible options that could be presented to the user are numerous. Many will enhance the visual interpretation of the problem. Some may never be called into use. Below is a list of possible options that could be included in this animation system:

(1) change window coordinates
(2) change viewport coordinates
(3) change scale factor
(4) change angle of rotation
(5) translate picture
(6) split screen into two parts
(7) interrupt processing of model
(8) change speed at which picture is moving
(9) change problem being analyzed

The window and viewport are two concepts which might not be understood by the average user of this system. As explained in the description of this system, this animation package has been developed to be used by engineers for visual aid in interpreting analytical results from various engineering problems. It has not been developed with the experienced graphics user in mind. Rather it is primarily for the inexperienced graphics user. Therefore the options to change the window or viewport coordinates will not be offered. If these options were to be offered, a lengthy explanation about viewport coordinates being a direct representation of the screen coordinates and window coordinates being a virtual representation of the screen coordinates would have to be included. Also how these two sets of coordinates interact with one another as far as the aspect ratio would have to explained. I feel the user would be more comfortable with options she more readily understands. This problem of understanding the options involved in a package seems to be one of the drawbacks in other animation system already designed. Many of them required an introductory course on the use of the system being utilized.[4] This animation package will hopefully eliminate the need for an introductory course explaining its use. I also think the less explanations or less text that has to be used on the screen the better the user will like it and the more she might be apt to use the system. This idea is suggested by Stephen Zwarg in an article in the AFIPS Conference Proceedings, 1972 entitled "Sailing - An Example of Computer Animation and Iconic Communication".[5]

The window and viewport coordinates will be set internally. The user will have control of the window and viewport through the options to scale the picture and to split the screen into two sections (discussed later). Of course the interaction between the options given the user and the two sets of coordinates described above will be totally transparent to the user.

Changing the scale of the picture on the screen is one option that will be given to the user. This will change the size of what is being viewed. The user will be asked to input two numbers, an X value and a Y value. These X and Y values will change the width and

---

4. S.M.Zwarg, "Sailing - An Example of Computer Animation and Iconic Comunication", AFIPS, Vol. 40, (1972), p. 1006.

5. Ibid., p. 1007.

the height respectively. The user can change one or both of the dimensions. The dimensions do not have to be the same. For example, if the user wanted to change the Y scale to increase the height by two, then when asked to input the scale factors she would input 1.,2.. 1. being associated with the X scale and 2. being associated with the Y scale. The option to scale was chosen over the options to change the window and viewport for the reasons stated previously. To further defend this choice, I asked a number of engineers who work with me which option they would feel more comfortable with for changing the size of a picture on a CRT, (1) change the scale factors X and Y or (2) change the viewport coordinates XMIN, XMAX, YMIN, YMAX. Only those people who had worked with graphics knew what I meant by viewport coordinates. The overwhelming response was for (1) - scaling.

When the scale is changed a certain point located in the system being viewed doesn't change position on the screen. The picture expands or contracts about this point. If the choice of the origin in the system was poor - not centrally located - a change of scale appears to move the picture on the screen as well.

The next option to be considered is rotation. The user will most likely be satisfied looking at her model straight on, in other words, exactly as her engineering problem constructed it. However sometimes, the user may want to view the picture at a different angle. This might occur if there is some problem with the engineering model and a rotation of the picture would help the user get a better perspective of the problem. The value to be input in this case is an angle in degrees that will be used to rotate the picture accordingly. If the user changes the angle of rotation several times and then wants to go back to the original position, she does not have to remember all the rotations inputted thus far. All that will be required of the user is to input an angle of 0.0 and the picture will be restored to its original position.

Translation may be needed if the picture has been clipped at one side of the screen. The user may not want to rescale the picture, but would rather keep its size and just move the picture so it's totally within the viewing screen. Here the option to translate the picture would come into use. Although this option may not be used very often, it is one that should be available if needed.

Another option that could be included is the idea of a split screen. Would a user like to view two active models, side-by-side? Would this be too distracting? If this was included, could the user control each model separately, having the ability to use any of the options discussed on either model? Certainly a split screen option would be an attractive feature. Many times engineers have problems that are similar in all but one variable. To compare two problems that are almost the same, but with one variation (i.e. two mass-spring damper systems with different loads) is often desirable and necessary. The question of distraction is taken into consideration. Therefore this option will only allow for a split screen. No more than two pictures will be able to be viewed at any one time. What about the option to control each model separately? This will also be allowed. The reason for inclusion of this option is explained with an example. Perhaps instead of two different models the user might want to view one model, the representation of which is on both sides of the split screen. Now she varies one side of the screen by changing its scale, angle of rotation, speed, etc. and compares this model with the original one on the other side of the screen. This would be a viable choice for some users.

Another option that is desirable is the interruption of the processing of the picture; in other words, suspending the motion of the model. At this point the user would be allowed to make a change in one or more of the options given, or terminate the session. If the user wishes to make a change she is given the choice of starting the model over with the new changes in effect, or continuing from where the picture was stopped with the new changes in effect. Of course, if the user wanted to stop the motion of the picture and just start over or continue without changing any options, this could be done also. This interruption process is something that most engineers would make use of regularly. This is much more desirable than having to wait to change an option until the motion of the picture is completed.

With this animation system, the user will be viewing her model moving in real-time according to the engineering problem being analyzed. There are times when it is desirable to see the model move at a slower rate, say 1/5 of real-time. The opposite is also true. There may be times when the user would like to view her model moving at a faster rate than real-time. There will be limits placed on how fast or how slow the motion may become. This is necessary to keep the quality of the picture intact. An appropriate message will be written to the screen so as to advise the user of the problem. This option will be very helpful in the study of many engineering problems. To slow down the model, interrupt the motion, then speed the movement back to real-time is the kind of control any user would want to see on a system designed for visual aid to engineering problems.

One final option would be in the interest of the user. That is the ability to execute the problem involved on the system, view the results in real-time, then execute another problem and repeat the process for as long as necessary. To do this kind of processing, this animation package would have to be linked to a large computer. This is because most engineering problems that would be executed in conjunction with this animation system take a lot of memory and a lot of CPU time. To have to sit and wait for the analysis to be complete before starting another model moving would not be very enjoyable. One alternative is to limit the types of engineering problems that could be executed on-line (on-line meaning with direct feedback to the terminal, rather than batch processing). The other alternative is to require that the engineering analysis be done by batch processing and the results needed for the animation package be stored on some disk file accessible by the animation program. The user should be given the types of analysis that can be done on-line in a relatively short period of time (up to sixty seconds). If what the engineer wants to analyze is not among the choices given, she must run her job batch and proceed as suggested in the second alternative.

All the options to be included are necessary to have a good interactive animation package. Although not all users will take advantage of all the options given, they should be included for occasional use.

HARDWARE AND SOFTWARE INVOLVED

Both the hardware and software used for this animation system are products of Tektronix, Inc. The computer used is a Tektronix Interdata 716 minicomputer. It is a 16 bit machine with 64 K bytes of core. There are four 4905 disk drives, each with five megabytes of storage. The CRT is a 4081 graphics tube with a 2048 x 2048 raster matrix. The Tektronix graphics software used is the Plot 80 DGSS (Distributed Graphics Support Subroutines).

Besides the Tektronix software, there must be a group of other programs that drive the graphics routines. This is a set of 14 FORTRAN routines written to utilize the Tektronix graphics routines to produce the animation package. These are described in the next chapter.

All the above software and hardware was involved in the animation system itself. There is some preliminary preparation needed before a user may run ENANIM that uses other software and hardware. Before running ENANIM the engineer must execute whatever analysis program whose results she is interested in displaying via ENANIM. This pre-animation programming must be done on the main computer facility at Homer Research Laboratories which is an IBM 3032. The results from the engineering analysis problem are sent to the Tektronix minicomputer via phone lines. This preliminary programming is necessary because the limited core size of the Tektronix computer used makes doing engineering calculations on it impractical.

ACTUAL SYSTEM DESIGN

The actual system design differs from the optimum system design described previously. One of the most obvious reasons is the need to consider the available software and hardware. As was specified in the previous chapter, Tektronix hardware and software was used for this animation system. The entire package was written in FORTRAN because the Tektronix graphics routines were designed to be called from FORTRAN programs. Before describing the routines written for this animation system I will discuss the graphics software utilized.

The pictures that are seen when viewing this animation system were created as a collection of entities called picture segments. A picture created as a picture segment can be displayed in either storage or refresh. A picture segment is a collection of vectors grouped together. The advantage of creating picture segments is that you can control each segment independently from all the others. Some segments may move across the screen while others remain stationary.

There are three steps to creating a picture segment:

(1) initializing the segment

(2) specifying the segment's vectors and vector attributes

(3) terminating segment creation

Initializing a segment amounts to one subroutine call - CALL OPEN(n), where n is an integer number identifying the segment. This number is always used to reference this segment. The segment's vectors can be specified by a variety of subroutine calls. Some of the ones used in this package follow: (1) CALL MOVE(x,y), where x and y are an x,y coordinate pair. This defines an invisible line to x,y from the last position specified. (2) CALL DRAW(x,y), where x and y are again a coordinate pair. This defines a visible line to x,y from the last position specified. (3) CALL CIRCLE(xy,radius,angle1,angle2), where xy is an array with two elements specifying an x,y coordinate which represents the center of the circle, and radius is obviously the radius of the circle. Depending on the application an entire circle or only part of a circle can be drawn. If an entire circle is desired, angle1 and angle2 can be omitted. If part of a circle is desired angle1 specifies the initial angle and angle2 specifies the final angle. An arc is drawn from angle1 to angle2 in the direction of the sign of angle2-angle1. (If the sign is positive, the arc is drawn counter-clockwise. If the sign is negative, the arc is drawn clockwise). (4) CALL SPLINE(xy,n,order,w), where xy is an array filled with n x,y coordinate pairs, order specifies the curve tension, and w is a work array used by SPLINE. There must be at least three x,y pairs in order to use SPLINE. By grouping these and other calls together in the appropriate manner, a specific picture is created. To terminate segment creation a call is made to subroutine CLOSE.

Once a segment has been created it can be displayed in storage or refresh and can be kept stationary or moved around on the screen. To display a picture segment in storage a call to FIX(n) is made with the segment number used as the argument. This picture is drawn once and will remain on the screen until the screen is erased. To display a picture segment in refresh a call to POST(n) is made with the segment number used as the argument. A refresh display is automatically drawn over and over again 30 to 60 times per second in order for it to remain visible. A call to UNPOST(n), DELETE(n) or CLEAR will erase a refresh picture. UNPOST(n) terminates the refresh display of segment n. DELETE(n) terminates the refresh display and frees the memory containing the segment definition of segment n. CLEAR terminates all pictures displayed in refresh and frees the memory containing all segment definitions.

Two types of movement are produced in this animation system. For one type, a segment is created, displayed in refresh, deleted, then created again with new vectors, displayed in refresh and deleted. This process is repeated as many times as necessary and gives the illusion of movement. The other type of movement involves defining the starting position of a segment, called its setpoint, and drawing the rest of the segment's vectors relative to this setpoint. First this segment is displayed in refresh. Then in order to move it, the setpoint is changed and the segment is displayed again in refresh. By continually altering the setpoint and redisplaying the segment in refresh the illusion of movement is obtained. This second method executes faster on the Tektronix computer because the segment doesn't have to be recreated. There are certain times when you would have to use the first method instead of the second method. For instance a diving board segment would have to be manipulated in the first method. The second method maintains the original shape of the segment created and moves it around on the screen. This is not the type of movement desired to represent a diving board vibrating. The entire board does not move, just one end of it. On the other hand a mouse jumping off a diving board is an example where the setpoint method would work best.

There are fourteen routines, besides the graphics routines described above, that make up ENANIM.[6] The main routine initializes the graphics environment, prints an introductory message and calls subroutines HEADNG and DRIVER. BLOCK DATA subroutine initializes some variables. CHECK is the routine that checks to see if the model is outside the window screen tolerance and prints an appropriate message when necessary. DRIVER is the routine which interacts with the user. Options can be changed in this routine and then it calls GRPHIC. ENGIN reads the engineering input data from disk into the proper time and displacement arrays. GRPHIC calls ENGIN to get the engineering data and then calls the animation routine, MICKEY. HALT is called when the user stops the motion of the picture. It also interacts with the user. Options can be changed or not changed and

---

6. These routines can be found on file in Christmas-Saucon, Rm. 102.

control is either returned to the animation routine or sent to GRPHIC or the session is terminated. HEADNG is another introductory routine which is called at the beginning of the session and prints out the options available for the first time. MESS prints out the message about the function keys option. MICKEY is the main animation routine. It calls all the routines that create picture segments and it creates the diving board segment. It then produces the animation movement using the engineering data received from ENGIN. MOUSE creates the mouse picture segment. TIMER creates and alters the time clock picture segment. WALL creates the wall picture segment and WATER creates the water picture segment.

Exactly how does this animation system work? There are four steps involved. The first two are done on the IBM 3032 and the last two are done on the Tektronix minicomputer.

(1) An engineer will run an analysis problem with the results (time,displacements and other information) stored on a disk file associated with the IBM 3032.

(2) The user executes another program which massages the results so they are in the proper form for ENANIM to accept.

(3) The user then executes a program that will receive the data from the 3032 and store it on a disk associated with the minicomputer.

(4) The user executes the animation package by typing in "RUN ENANIM".

The reason for the many steps was stated in the previous chapter. The minicomputer is too small in core size to execute an engineering analysis problem. Therefore that must be done first on the main computer which is an IBM 3032.

Presently ENANIM is configured so that every time step has one displacement value associated with it. Therefore the disk files created in steps (2) and (3) consist of two values for each record in the files. In more advanced analysis there may be more displacements per time step. ENANIM could be altered to accept this configuration. Currently it is not.

The user interacts with ENANIM in two ways. Both the keyboard and the functions keys can be utilized. When responding to a question posed, the user answers by depressing the appropriate keys on the keyboard. The function keys are provided in case the user wishes to stop the motion of the model. If she does want to halt the picture, she just has to depress any one of the twelve function keys and motion is suspended. The user may go in many directions at this point. She can change an option and start the model over again, she can start the model over without changing anything, she can continue from where the model was halted or she can terminate the session.

There are basically two types of screen displays used in this system. There are menus, questions posed and other written material; and there is the animation model itself. When viewing the animation model, the user sees the model moving in the center of the screen. A time clock is displayed in the upper right-hand corner of the screen and a message about the function keys option is displayed in the upper left-hand corner of the screen. When the menu is displayed it contains all the transformation options offered to the user. These include scaling, rotating or translating the model. Another option offered in the menu is changing the speed of motion of the model. These options plus the function keys option described above constitute all the options allowed by ENANIM. The questions asked are all stated simply and clearly so the user will have no problem answering them. If the user keys in an inappropriate response a message is printed saying that the input was not valid. The program then allows the user to try again. If the user changes the scale factor, the angle of rotation or the translation coordinates so the picture is off the screen, clipping will occur. No message is printed because the user will visually see the result. If the user changes the speed so it is too fast for the computer to handle a message is printed and speed is set back to real-time.

COMPARISON WITH EXISTING ANIMATION SYSTEMS

Since the advent of computer graphics systems there have been animation packages developed with many different purposes, specialties and output mediums. Some are very complex and very general while others are simpler and more specialized. In order to evaluate the computer graphics animation system presented in this thesis, a number of other animation systems were examined. Each will be discussed separately and its similarities to and differences from to ENANIM will be explored.

ANTICS(3) is a system developed at the University of British Columbia, Vancouver, Canada for animating the execution of LISP programs for educational purposes. The system is implemented on a Model 10 Adage Graphics Terminal which communicates with an IBM 370/168 computer. The programs that ANTICS is comprised of are user written subroutines interspersed with calls to animation routines. ANTICS has provisions for producing animated films, film strips or slides depicting the execution of LISP programs. It can also be used interactively so a student can sit down and have a lesson with ANTICS concerning LISP. The human factors that the developers of ANTICS were concerned with were the following:

(1) graphical representation of concepts

(2) timing and sequencing of animation

(3) response time and interaction methods.

With ANTICS, the display on the screen may be split into several sections. The definition of a function in LISP may be displayed at the top of the screen, with that portion intensified that is currently being animated. The LISP stack or association list may also be displayed on the screen if the user wants to see the changes to these as they occur. As the screen fills up and the text reaches the border of the display area, part of the display is "rolled off" the screen in order to free space on the screen. There is usually a lot of text on the screen at any one time. This is because ANTICS is a system for animating the execution of LISP programs and therefore very few pictures are involved. The "rolling off" feature works well for this animation system, but is unnecessary for ENANIM. The screen in my system is more apt to be empty than full. ENANIM was designed this way to limit distraction for the user.

The overall speed of the animation in ANTICS had to be slow enough for users to follow. This is especially true because of the amount of text appearing on the screen at any one time. With ANTICS, the speed of the animation can be adjusted according to the viewer's skill. The graphical input devices that ANTICS uses are the light pen and function keys.

Like ENANIM, ANTICS was developed with a specific purpose in mind. They are both used interactively as an instructional tool. It appears that ANTICS is geared toward the inexperienced user as is ENANIM, although ANTICS does require some training on the part of the user. A major difference between the two animation systems is that ENANIM can only be used interactively while ANTICS can also produce an animated film. Another difference between the two systems is the animation technique employed.

ANTICS uses "key frame" animation technique while ENANIM uses "direct" animation technique. In the key frame technique an image is produced for every "key" frame of the finished product, and the computer produces intermediate interpolating frames, resulting in a smooth movement of the image from one key frame to the next. ENANIM uses the direct technique because of the nature of the graphics system (Tektronix) used. The Tektronix graphics system provides the means to scale, rotate, translate, etc. a set of vectors. These facilities are used to simulate movement and other animation effects. The different techniques are chosen depending on the primary applications involved and the hardware being used. The key frame method used by ANTICS is better for producing an animated film from a computer, while the direct method is better for interactive graphics systems.

Just as ANTICS allows the user control over the speed of the animation, so does ENANIM. Where this option is used in ANTICS to help the user understand the execution of a LISP program, it is used in ENANIM to help the user understand a model's characteristics as the model changes with time. ENANIM only lets the user control the keyboard and function keys, not the light pen as in ANTICS. In the optimum design of an animation system like ENANIM the use of the joyswitch would be included.

Another animation system that was investigated and compared to

ENANIM is one that was developed at Johns Hopkins University(10). Although at the outset this system seems entirely different from ENANIM, it is in fact very similar. The Johns Hopkins University animation system is used solely to produce animated film. It is not an on-line system, as there was no on-line graphics equipment available to the creators at the time of development. A second generation IBM 7094 computer was used. The animation system makes use of the language MOGUL - Movie Oriented Graphical Utility Language. It is Fortran-based as is the Tektronix software used for ENANIM. The animation package consists of three sections:

(1) DRAW - This is a set of drawing functions which enable the animator to create and manipulate images.

(2) MOGI - This is the executive portion which keeps track of the frame numbers and screen dimensions. MOGUL allows the user to define the viewport as a 2-D real plane. This section windows any coordinates outside this area.

(3) MOVIE - This is the device driver section which translates the normalized coordinates produced from MOGI into coordinates necessary to produce visual images on film.

Although the output media of ENANIM and MOGUL are different, the considerations that had to be made for each system are the same. One of the greatest concerns of Stephen Zwarg, an animator using MOGUL to produce a sailing film, was what symbols and images should be used to express ideas. He chose to minimize the use of text and concentrate on pictoral representations. The same problem was considered when ENANIM was designed and, except for the menu, it was decided to use only pictoral representations of the model being analyzed. What about the layout of images, and the occupancy of the screen? Zwarg chose to produce the minimum number of images, centered on the screen. The optimum animation system design disussed earlier allows the user a little flexibility here. The user can do as Zwarg did and view one model in the middle of the CRT, or the user can split the screen and view two models. Although Zwarg feels that a split screen detracts from the overall picture, in the case of ENANIM it could be very useful. As my animation system was specifically designed for visual aid for engineers, there are times when it would help the user's understanding of a particular problem if two models could be viewed simultaneously with some slight variation in one.

MOGUL has many of the same options incorporated into it as ENANIM does. The user can scale, rotate, translate and vary the speed of the animation. Where ENANIM lets the user do this in real-time, the animator using MOGUL must preprogram the options. Both systems allow a pause in activity to facilitate the understanding of difficult material. There is an obvious drawback to the animation system at Johns Hopkins. Because no immediate feedback is available the user of MOGUL must produce a film, view it, and then go through the whole process again if something is not right. With ENANIM there is also a looping procedure, but the part that must be done over is the analysis of the engineering problem involved. This is done apart from ENANIM while the part that must be repeated with MOGUL is the animation procedure itself. In the latter system the user actually programs the animation wanted. Therefore a knowledge of FORTRAN is required. In the former system the user just views the animation wanted. No knowledge of graphics or FORTRAN is required.

GENESYS(2) is an animation system developed at M.I.T. Lincoln Laboratories and used on its TX-2 computer. GENESYS stands for GENEralized-cel[sic] animation SYStem. This system is an interactive computer-mediated animation system. This phrase was coined by R.M. Baecker who did his Ph.D. dissertation on animation at M.I.T. GENESYS has the following capabilities:

(1) immediate feedback of results, final or intermediate.

(2) ability to factor picture construction into stages and view results after each stage.

(3) ability to sketch pictures directly into the computer.

By reading these three capabilities one can easily deduce that GENESYS is a far more sophisticated animation system than ENANIM. GENESYS allows the user to sketch key frames called cels on a tablet, and the picture is automatically displayed on the screen. These cels combined with global dynamic descriptions that the user inputs produce visible picture change. This is done by algorithms embedded in the animation system. Because of the sophisticated nature of this system it is necessary for the user to understand the language used in GENESYS. Some schooling may be necessary before the user could generate any pictures. There is no similar problem with ENANIM. As has been noted earlier, ENANIM was designed with the inexperienced user in mind.

GENESYS has the ability to allow the user to sketch directly into the computer. Therefore the user can alter the shape of the picture being displayed. ENANIM also lets the user alter the shape of the picture. The difference lies in the technique. ENANIM requires the user to specify which change she wants to take place - scaling, rotating, etc. by hitting the appropriate key on the keyboard. This is the same technique ENANIM uses for changing the speed of movement of the picture. GENESYS also allows the altering of speed of motion. As with the changing of the picture's shape, this is done with a tablet and light pen.

The free form sketching that is part of GENESYS is useful when one knows the general shape and quality of motion rather than the analytical expression for a function that determines it. ENANIM always and only uses analytical expressions for functions that determine motion because this system is specifically tailored for engineers to view analytical results change with time.

GENESYS is ultimately used to produce an animated film. For this purpose immediate feedback is an excellent component of this animation system. This option makes GENESYS that much more versatile than the system from Johns Hopkins that was discussed prior to this. Because of the nature of ENANIM, immediate feedback would be impossible. Immediate feedback in this case means being able to execute the engineering analysis on-line while the user waits to view it. Where this is an advantage to GENESYS, it would be a hinderance to ENANIM. This is from a human factors point of view. As it was discussed in an earlier chapter an engineering problem takes too long to execute (one minute to several hours) for a user to sit through the execution of a problem, then view it using ENANIM, and then try again with a variation of the problem.

As one can discern from the discussion above, GENESYS is a general purpose animation system used for producing films. ENANIM is a more rigid animation system used for one specific application.

Another animation system to be discussed was developed at Cornell Aeronautical Laboratory, Inc., Buffalo, N.Y.(9). There they were concerned with the pictoral representation of automobile dynamics. With a complex simulation such as automobile dynamics where there is a lot of interaction between components, the equations of motion become long and numerous. Besides this problem the output from a simulation run may be quite extensive and hard to understand. Like the problems facing the engineers who needed a system like ENANIM, this problem was diminished by creating a graphical display of an automobile moving down a road. Now a user could actually see the movement of a car and the interaction of its components rather than try to understand all of this be looking at a printout.

This system does not allow for on-line viewing of the movement of a car. The user has two output mediums to choose from. One type of output is to view the car in separate frames at different intervals in time. The second type of output is a film produced with the car moving in time. The entire simulation was done in 3-D with no hidden lines removed. Like ENANIM, this system employs clipping. If part of an object is outside the picture boundary then that part is not seen, but the rest of the object is. There is a minimum amount of text seen on the screen in this system as in ENANIM. The reasons for this are the same also: what is being viewed is explanation enough. Also in both systems the user who is viewing the picture knows what she is seeing because some analysis must be done prior to the animation.

This system allows the user to vary the speed of movement. The user can also scale or translate the picture to make sure the car stays in the middle of the screen. All these options are changed prior to the execution of the simulation. Therefore the user has no direct interaction with the system. In much the same way as Stephen Zwarg produced his sailing film using MOGUL, this system produces a film. The user constructs the car with certain parameters specified (position of sprung mass, camber of each front wheel, deflection of the rear axle, etc.), indicates animation options (speed, scaling, etc.) and then runs the program. This process must be repeated if any alterations are desired. This process is also similar to ENANIM. In fact ENANIM is more similar to this system than to MOGUL because both this system and ENANIM must have some computing analysis done before the animation can be viewed. The difference lies in the fact that ENANIM has the user define the animation options interactively while this system has the user define the options before the problem is executed.

It has always been difficult to teach an intuitive approach to subjects such as physics. Computer animation provides a means for modeling phenomena of nature which can help students visualize some effects more directly than by looking at equations. Because of a computer's ability to produce graphical displays the M.I.T. Education Research Center began looking into this as a means to help students better understand certain laws of physics. The paper(6) written by Judah Schwartz and Edwin Taylor discusses attempts to present visually and directly as possible some results on certain topics in physics using computer displays.

A student can sit in front of a CRT, and by manipulating the display can become familiar with effects due to time delays for light propagation and effects due to the kinematics of spacetime. As in ENANIM, this system has very basic displays with little text. In one example the screen shows a road with telephone poles down both sides. The user controls a "rocket" car's forward and backward acceleration down the road by using a light pen. The telephone poles are just lines and the user never sees the car. Instead the poles bend depending on the car's speed. The only text seen is the speed of the car measured as a fraction of the speed of light, the time in arbitrary units measured in the cockpit of the car, and time in the same units read off a series of road clocks.

Another display helps a student understand the kinematics of spacetime. Here the user has a few more options than in the first example. The student uses a light pen and touches the screen to select an option. This is the same method the student in the first example chose to accelerate or decelerate the rocket car. In an optimum design perhaps using the joyswitch to control forward and backward acceleration might be better. This technique of touching the screen with the light pen is really no different than pressing a key at the keyboard (as in ENANIM). With this example the user has the options to erase, reset, scale or transform the display. This is due to the nature of the example which sets an initial frame of reference (zero time and space) and then allows the user to transform the picture to another frame of reference. The relative velocity between frames increases continually as long as the light pen is pressed against a certain spot on the screen. Reverse transformation back to coordinates in the original reference frame is also possible. This is used to teach a variety of spacetime properties (Doppler shift, time dilation, etc.).

The fact that this system is used for visual aid in understanding some very complicated laws of physics identifies it as similar in purpose to ENANIM. ENANIM has more text appearing on the screen for choosing animation options. Because this system at M.I.T. does not explain the options on the screen, the student must know about them before executing any problem. This will speed up the session on the CRT, but some preparation beforehand will be necessary.

EVALUATIÓN OF THE SYSTEM

In order to evaluate ENANIM a look must be taken at the limitations of the system. These limitations are based on a comparison between the optimum system design and ENANIM. An evaluation is also made when considering ENANIM on its own merit.

There were a number of design options in the optimum animation system that were not included in the actual animation system implemented. These include the option to split the screen into two sections so the user can view two problems simultaneously, the joy-switch control to increase or decrease the speed of the model, and the option to use single keys for particular commands such as 'S' for stop or 'C' for continue. The exclusion of these options limited the versatility of the ENANIM.

Another major limiting factor to this animation system is the small size of the computer that was available for ENANIM. This forced the users of ENANIM to execute their engineering problems off-line on another, much larger, computer. If a user finds a problem in the analysis or wants to vary some aspect of the analysis after running ENANIM, she must rerun the engineering problem, store the results, access the results, and then run ENANIM again. This delay is caused by the length of time and amount of core that is needed to run an engineering problem. In the ideal situation, there would be a direct link between the minicomputer running ENANIM and the main computer executing the engineering problems. This could eliminate much of the lag time.

Another problem associated with the size of the computer is the possible size of the animation program. On a 64 K byte machine, taking into consideration the operating system and overhead, the amount of storage available to ENANIM is approximately 30 K. To make any major design changes, perhaps to tailor ENANIM more to the optimum system design, more than 30 K bytes of core is needed. Besides the fact that the engineering problems have to be run off-line, here is another reason why the size of the Tektronix minicomputer is a limiting factor.

As ENANIM stands now, is it really a viable tool which can benefit an engineer in analyzing a problem? After running ENANIM many times, I can see that there are a few areas which need to be improved in order for ENANIM to really be a workable animation system.

Currently ENANIM is limited in the amount of engineering input data that it can accept and act on. ENANIM allows one displacement value for every time step. There are many engineering problems where there would be multiple displacements at each time step. By limiting the input for ENANIM, you are limiting the number of applications that can take advantage of a real-time animation system like ENANIM. This is a section of ENANIM that could be altered so multiple displacements per time step could be used.

Another problem lies in the number of time and displacement values that ENANIM uses. If the function displaced over time is very uneven (rapidly changing values in a short period of time followed by a fairly smooth section where the function changes very little), the time increments must be very small so all change can be seen. If the time steps used are too large ENANIM will not be able to accurately display the function in question. The time increment used is determined be the engineering analysis problem. Therefore it is up to the engineer to choose the proper time increment. Because this is not always possible the above problem will occur.

Presently ENANIM has picture segments that are able to depict simple problems of vibrating beams or mass-spring damper systems. If ENANIM is to be used for any engineering problem, many more picture segments would have to be created. The type of engineering problem would determine the picture segments used. As explained above, the core size limitation hinders this improvement.

Now that ENANIM's limitations, both compared with the optimum design system and evaluated on its own merit, have been discussed, what conclusion can be drawn? Is ENANIM a worthwhile animation system? Will it be used?

For an engineer who wants to see an object move with time, ENANIM is an easy-to-use, easy-to-understand tool. The menus displayed are easy to read and can be followed with little trouble. The options are adequate for an animation system. The ability to stop the motion of the model and start it again can be very helpful. The fact that an engineer will have to wait some time between the problems that are executed on ENANIM is something that she is already used to. Engineering problems have always taken a long time to run. On the other hand, there are the problems involved with the input data, both the number of displacements per time step and the total number of time and displacement values, the number of picture segments available and the size of the Tektronix minicomputer. Taking all the above into consideration, ENANIM is a tool which can be used for simple engineering problems, but for more advanced engineering problems some major modifications would have to be made.

BIBLIOGRAPHY

1. Baecker, R.M. Interactive Computer-Mediated Animation. Massachusetts: M.I.T - Project Mac, June 1969.

2. Baecker, R.M., "Picture-Driven Animation", Proc. American Federation of Information Processing Societies (AFIPS), Vol. 34 (1969), pp. 273-288.

3. Dionne, M.S. and A.K. Mackworth, "ANTICS: A System for Animating LISP Programs", Computer Graphics and Image Processing, Vol.7, No. 1 (Feb. 1978), pp. 105-119.

4. Foley, James D. and Andries van Dam. Interactive Computer Graphics. Massachusetts: Addison-Wesley Publishing Co., Inc., 1979.

5. Knowlton, K.C., "A Computer Technique for the Production of Animated Movies", Proc. AFIPS, Vol. 25 (1964), pp. 67-87.

6. Schwartz, J.L. and E.F. Taylor, "Computer Displays in the Teaching of Physics", Proc. AFIPS, Vol. 33 (1968), pp. 1285-1292.

7. Talbot, P.A., J.W. Car, III, R.R. Coulter, Jr. and R.C. Huang, "Animator: an On-line Two Dimensional Film Animation System", Communications of the Association for Computing Machinery, Vol. 14, No. 4 (April 1971), pp. 251-259.

8. Tektronix, Inc. Plot80 DGSS (Distributed Graphics Support Subroutines) Reference Manual. 2 vols. Oregon: Tektronix, Inc., 1978.

9. Theiss, C.M., "Computer Graphics Displays of Simulated Automobile Dynamics", Proc. AFIPS, Vol. 34 (1969), pp.289-296.

10. Zwarg, S.M., "Sailing - An Example of Computer Animation and Iconic Communication", Proc. AFIPS, Vol. 40 (1972), pp. 1005-1014.

VITA

Suzanne Davis Vitale was born to Louise and William Davis on January 19, 1952 in Orange, New Jersey. After receiving her undergraduate degree in mathematics from Cedar Crest College, Allentown, Pa. she worked at Bell Telephone Laboratories, Holmdel, N.J. for 2 1/2 years. It was at Rutgers University in New Brunswick, N.J. that she began her M.S. in Computer Science. Suzanne then moved to Whitehall, Pa. and took a job as a scientific programmer/analyst for Bethlehem Steel Corp., Homer Research Laboratories, Bethlehem, Pa. She was able to transfer her Rutger's credits to Lehigh University and completed her M.S. degree in Computer Science at Lehigh.